原因と対策の説明

1. **原因箇所**

以下の関数の赤字(ループ回数を設定する所です)

def create\_dataset(dataset, look\_back=1):

dataX, dataY = [], []

**for i in range(len(dataset)-look\_back-1):**

xset = []

for j in range(dataset.shape[1]):

a = dataset[i:(i+look\_back), j]

xset.append(a)

dataY.append(dataset[i + look\_back, ])

dataX.append(xset)

return np.array(dataX), np.array(dataY)

1. **原因説明**

今回の設定では、len(dataset)=20, look\_back = 8になっていますので, len(dataset) - look\_back – 1 = 11になるので、for i in range(len(dataset)-look\_back-1)のループでは11回のループになります。

　Datasetの長さは20なので、datasetの値はt0,t1,…,t19とします。そうすると、create\_datasetを実行する結果、dataXの結果は以下のようになります

　[[t0-t7],

[t1-t8],

[t10-t17]

]

このdataXを使って、予測すると、次のタイムスタンプを予測するので、t18に近い値を予測します。

　この後、dataXの先端の値を削除して、予測データをdataXの最後に入れると、２回目の予測で使うDatasetは以下のようなイメージになります。

t1,t2,…,t19,t18‘(予測結果はt18に近い値になるので、t18’の記号を使います)。そうすると、datasetの規則が崩れ、予測値の結果は期待したものにはなっていないと考えられます。

1. **対策案**

今回の要望”現時点での最新データを使い予測値を取得し、その値を使い”というものであれば、datasetの値はt0,t1,…,t19の場合、create\_datasetを実行する時の戻り値　dataXは以下のようになる必要があります。

　[[t0-t7],

[t1-t8],

[t10-t17],

[t11-t18],

[t12-t19]

]

そうすると、予測値は次のタイムスタンプt20に近い値を予測してくれますので、要望に対応できる形になります。

1. **対応ソース　（赤字は対応する所です）**

def create\_dataset(dataset, look\_back=1):

dataX, dataY = [], []

**for i in range(len(dataset)-look\_back+1):**

xset = []

for j in range(dataset.shape[1]):

a = dataset[i:(i+look\_back), j]

xset.append(a)

//dataY.append(dataset[i + look\_back, ]) #配列のindex外にアクセスするので、エラーになるため、ここはコメントアウト。予測では、dataYを使っていないため、影響がない

dataX.append(xset)

return np.array(dataX), np.array(dataY)